Matching MEDLINE/PubMed Data with Web of Science (WoS): A Routine in *R* language

Daniele Rotolo[*1] and **Loet Leydesdorff**[†2]

[1]SPRU — Science Policy Research Unit, University of Sussex

[2]Amsterdam School of Communication Research (ASCoR), University of Amsterdam

Brief communication — Version: July 10, 2014

Accepted for publication in the

Journal of the Association for Information Science and Technology

Abstract

We present a novel routine, namely *medlineR*, based on *R* language, that enables the user to match data from MEDLINE/PubMed with records indexed in the ISI Web of Science (WoS) database. The matching allows exploiting the rich and controlled vocabulary of Medical Subject Headings (MeSH) of MEDLINE/PubMed with additional fields of WoS. The integration provides data (e.g. citation data, list of cited reference, list of the addresses of authors' host organisations, WoS subject categories) to perform a variety of scientometric analyses. This brief communication describes *medlineR*, the methodology on which it relies, and the steps the user should follow to perform the matching across the two databases. In order to specify the differences from Leydesdorff and Opthof (2013), we conclude the brief communication by testing the routine on the case of the "Burgada Syndrome".

Keywords: Medical Subject Headings; MEDLINE/PubMed; Web of Science; database integration; *R*; case-study.

[*]Corresponding author: d.rotolo@sussex.ac.uk, Phone: +44 1273 872980

[†]loet@leydesdorff.net

1 Introduction

The use of Medical Subject Headings (MeSH) of the US National Library of Medicine (NLM) for scientometric analysis of the medical context has increased in the last few years. These include the use of MeSH terms to delineate domains (Lundberg et al., 2006), to identify emerging topics (Ohniwa et al., 2010) and research areas (Guo et al., 2011), to map the dynamics of emerging technologies (Leydesdorff et al., 2012), to evaluate the impact of funding sources on the number of citations publications receive (Boyack, 2004), to perform co-word analyses (Stegmann and Grohmann, 2003), and to disambiguate author names (Torvik et al., 2005).

The MeSH classification, which is integrated in MEDLINE/PubMed, is a rich and controlled vocabulary generated through an intense indexing process performed by examiners. Terms, namely 'descriptors', are assigned to documents to delineate their content at different levels of specificity. The 2014 MeSH vocabulary is specifically composed by 27,149 descriptors which are organised in a tree-like structure.[1] Descriptors may be also complemented with one or more 'qualifiers'. These terms further contextualise the meaning of the descriptors to which they are assigned in relation to the content of the considered document.

The rich vocabulary provided by the MeSH classification can be used to delineate samples of documents in a number of medical areas and, as discussed, at different levels of specificity. For example, if one aims to examine the publication activity associated with a given disease, the MeSH descriptors associated with this disease can be identified in the "MeSH browser interface"[2] and used to build a search string in the MEDLINE/PubMed interface. The retrieval of the set of associated documents is consequential. Using MeSH-based search strategy has been suggested as an approach preferable to using keywords in titles (and abstracts), journals, or authors' names for the delineation (Lundberg et al., 2006).

Data from MEDLINE/PubMed however poses important limitations to scientometric analyses since they do not include key fields that are instead listed in commercial databases as ISI Web of Science (WoS) and SCOPUS. These include a publication's number of citations, list of cited references, as well as list of the addresses of authors' host organisations. For example, MEDLINE/PubMed only provides corresponding authors' addresses, while full information is

[1] The MeSH tree is organised in 16 branches representing different medical areas (e.g. diseases, chemical and drugs, therapeutic techniques). Additional details on the MeSH tree are available at `ww.nlm.nih.gov/pubs/factsheets/mesh.html`

[2] `www.nlm.nih.gov/mesh/MBrowser.html`

required to investigate co-authorship networks at organisational level, to perform geographical mapping, and to aid the disambiguation of authors' names.[3]

WoS and SCOPUS databases integrate the MeSH classification. WoS has a dedicated MEDLINE interface, while SCOPUS enables searching MeSH descriptors within the "Indexterms()" field. However, "Indexterms()" is not specific for the MeSH classification. These terms are assigned to SCOPUS records by indexers and integrated with thesauri that Elsevier owns of licenses. Searches within the "Indexterms()" field therefore extends also to records not included in MEDLINE/PubMed. WoS MEDLINE interface also has some limitations. It does not provide direct access to the above-listed key fields, i.e. the MEDLINE interface is 'weakly' integrated with the WoS core interface. The user has, for example, to enter record by record in WoS MEDLINE interface to collect citation data since they are not included in the download of the identified set of documents (based on WoS 5.13 version).

Following the lead of previous works (Leydesdorff and Opthof, 2013), this brief communication presents a novel routine, namely *medlineR*, which enables the integration of data obtained by querying MEDLINE/PubMed with data available in WoS. The routine builds on some of the basic functions included in *R*, a widely diffused open-source language and environment available across multiple platforms, and on the package "*stringr*" (Wickham, 2010), which can be easily installed in the *R* environment by the user.

The *medlineR* routine and methodological approach is described in the followings. We apply the routine to the same case-study Leydesdorff and Opthof (2013) investigated. This allows specifying differences between the two routines. We conclude the brief communication discussing the implications deriving from the integration of data across MEDLINE/PubMed and WoS.

2 Methods

To match data collected from MEDLINE/PubMed with WoS data by using *medlineR*, the user has to follow a number of steps that are listed below. The code is reported in the Appendix and the *medlineR* script is available at `http://www.danielerotolo.com/#!medliner/cid7`.

1. The user first installs *R*[4] and the package "*stringr*" (Wickham, 2010).[5] The latter can be

[3] The NLM has recently announced a new policy to enable publishers to submit full information on authors' addresses starting from October 2013 (NLM Tech Bull. 2013 Sep-Oct;(394):b4).

[4] *R* can be downloaded at `www.r-project.org`

[5] Details on the package are available at `http://cran.r-project.org/web/packages/stringr/index.html`

installed from the *R* command line using the following string *install.packages("stringr")*. An Internet connection is required.

2. The user can identify a list of "PubMed Identifiers", namely PMIDs, in MEDLINE/PubMed to match with WoS data. The list of PMID can be derived from the set of publications obtained from the specific query the user is performing. The interface of MEDLINE/PubMed allows for the download of PMIDs in *txt* format.[6] It is worth noting that *medlineR* can also work directly at WoS MEDLINE with a valid search string in the advanced-search interface. In this case, the user skips step 2 and step 3.

3. Assuming that the list is composed by M PMIDs, the user builds an advanced search string in the WoS MEDLINE interface according to the following syntax: $PM = PMID_1$ *or* $PM = PMID_2$ *or* ... $PM = PMID_M$. Conventional spreadsheets can be used to generate the search string.

4. This search string is then used to query WoS MEDLINE through the advanced search interface. The query will return a list of documents. Each document can be accessed through a weblink. This implies that a *url* link is associated to each document the search retrieved.

5. The user has to input three parameters in the *medlineR* script:

   (a) One of the *url* links to WoS MEDLINE documents (variable *wosurl*). To do so, the user can access to the first document in the list and copy-paste, between quotation marks, the *url* link associated with this document in the *R* code — *medlineR* will use this *url* to generate the remaining ones automatically.[7]

   (b) The number of documents to collect (variable *numdocs*).

   (c) The path to the folder in which the outputs of the *medlineR* routine should be saved. This should be inputted in *R* between quotation marks (e.g. "C:\\Users \\user" in

[6] `www.ncbi.nlm.nih.gov/pubmed`

[7] If the user builds a search string in the advanced search interface of WoS MEDLINE and the search returns more than 10 documents, the *url* associated to one of these documents will have the following format: "`http://apps.webofknowledge.com/full_record.do?product=MEDLINE&search_mode=AdvancedSearch&qid=***&SID=***&page=***&doc=***`". Stars replace codes that are generated by WoS MEDLINE according to the performed search and the session number. In the case of a search in the general interface the *url* will have the following format: "`http://apps.webofknowledge.com/full_record.do?product=MEDLINE&search_mode=GeneralSearch&qid=***&SID=***&page=***&doc=***`".

Windows or "/Users/username/Desktop/" in Mac OS X).[8]

6. The user has to launch *medlineR* (the shortcut in Windows is "CTRL+A" and then "CTRL+R" whereas in Mac OS X is "CMD+A" and then "CMD+Return"), which parses the *wosurl* variable to generate the whole set of links pointing to the identified documents and collect the full *html* code of the associated webpages. The *html* code is then parsed to retrieve documents' UTs, i.e. "Unique Article Identifiers" in WoS. An indication of the number of processed records is reported in the *R* interface as the routine advances.

7. The *medlineR* routine generates three different outputs: (i) a set of files (sequentially named *wosPMID=1.txt, wosPMID=2.txt, etc.*) including the *html* code of each webpage, (ii) a document, called *wosut.txt*, which lists PMIDs with associated (when available) UTs, and (iii) a file, called *search.txt*, that provides the full search string for WoS as generated from the collected UTs.[9] This string can be use in the advanced interface of WoS to retrieve the full records of the identified documents. These can be then downloaded from the WoS interface.

3 A case-study: The Brugada Syndrome

For a comparative analysis with the results obtained by Leydesdorff and Opthof (2013), we applied the *medlineR* routine on the case-study of the Brugada Syndrome (BRs) that is a rare cardiac disease — for more details on the case-study see Leydesdorff and Opthof (2013). BRs is identified in the MeSH classification with the term "Brugada Syndrome", which is coded in the tree with "C14.280.067.322" as a cardiac arrhythmia and "C16.320.100" as a congenital disease. This term has been indexed since 2007.

We searched for all publications to which the "Brugada Syndrome" MeSH term was assigned during the 2010-2011 period. We performed the search in MEDLINE/PubMed on 6 June 2014 and compared the results obtained from WoS MEDLINE interface and SCOPUS by using the same search approach. Table 1 summarises the results. MEDLINE/PubMed returned 349 records while WoS MEDLINE and SCOPUS returned lower and higher number of records, respectively. This shows evidence of the limitations of applying MeSH-based searches directly on

[8] *R* requires double backslash when the path to the selected folder is specified in Windows.

[9] The *medlineR* routine may return warning messages at the end of the data collection process. The user can ignored these messages since they do not affect the produced files.

Table 1: Comparing databases: Records related to the "Brugada Syndrome" (2010-2011 period).

Database	**Search string**	**Records**
MEDLINE/PubMed	"Brugada syndrome"[MeSH Terms] AND ("2010.01.01"[PDAT]: "2011.31.12"[PDAT])	349
MEDLINE WoS	MH="Brugada Syndrome" AND (PY=2010 OR PY=2011)	323
SCOPUS	INDEXTERMS("Brugada Syndrome") AND ((PUBYEAR = 2010) OR (PUBYEAR = 2011))	591

Note: The searches were performed on 6th June 2014.

those databases.

We launched *medlineR* to match the records obtained from the search performed in MEDLINE/PubMed with those that are listed in WoS MEDLINE and then in the core interface of WoS. The routine identified 294 UTs out the 349 records included in the sample. As a comparison, we also used the *medline.exe* routine developed by Leydesdorff and Opthof (2013) the same day we used *medlineR*. The number of retrieved UTs was identical.

The *medlineR* routine does not requires large computational power or memory since most of the collected data are written on the hard drive as they are collected. The main parameters that affect its performance are the number of records to collect and the quality of the Internet connection. We used the routine on a Macbook Pro with a 2.8 GHz Intel Core i7 processor and 8GB (1333MHz DDR3) RAM. The data collection for the BRs case-study was achieved in about 11 minutes.

4 Conclusions

The rich and controlled vocabulary of the MeSH classification of MEDLINE/PubMed allows for rapid delineation of publication samples in medical areas at different levels of specificity. More than 27,000 terms (descriptors) populate the classification and the list of those is also constantly updated with new terms to cover emerging areas. MeSH therefore provides a valuable alternative to searches of keywords in the titles and abstracts of publications or to searches that relies on sets of journal titles. However, MEDLINE/PubMed data poses limitations to the scientometric analysis. Key fields required for scientometrics are missed or incomplete. For example, MEDLINE/PubMed does not list the references a publication cited neither future citations, which in turn are key inputs for bibliographic coupling and citations analyses, respectively. In addition, authors' addresses are reported only for the corresponding author, thus limiting the possibility to

examine inter-organisational networks, to perform geographical mapping, as well as to support the disambiguation of author names. The missing data also does not allow to perform scientometric mapping with a number of available mapping tools (e.g. the map of science based on WoS subject categories, the journal mapping) (for an overview, see Rotolo et al., 2014).

These fields are available in commercial databases as WoS and SCOPUS, which also include interfaces to build MeSH-based searches to retrieve publication data. Yet, in the case of WoS, searches are not comprehensive whereas, in the case of SCOPUS, searches extend also to documents not included in MEDLINE/PubMed. In this brief communication, we proposed a novel routine in *R* language, namely *medlineR*, that enables users to integrate, by using the WoS MEDLINE interface, data from MEDLINE/PubMed with the above mentioned fields available in WOS. *MedlineR* specifically matches PMIDs from MEDLINE/PubMed with UTs from WoS. The *R* language on which *medlineR* is based also enables the user to introduce additional functionalities such the parsing and retrieval of other fields available in the original *html* code of WoS or rapid analyses of the collected data directly in the *R* environment. The code is indeed editable and adaptable to the specific requirements of the scientometric analysis the user aims to undertake.

Acknowledgements

We acknowledge the support of the UK Economic and Social Research Council (award RES-360-25-0076 - "Mapping the Dynamics of Emergent Technologies"). Daniele Rotolo also acknowledges the support of the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) (award PIOF-GA-2012-331107 - "NET-GENESIS: Network Micro-Dynamics in Emerging Technologies"). We are grateful to Carlos Benito, Francois Perruchas, and Ismael Rafols for their feedback.

References

Boyack, K. W. (2004). Mapping knowledge domains: characterizing PNAS. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl(suppl_1):5192–9.

Guo, H., Weingart, S., and Börner, K. (2011). Mixed-indicators model for identifying emerging research areas. *Scientometrics*, 89(1):421–435.

Leydesdorff, L. and Opthof, T. (2013). Citation analysis using the Medline database at the Web of Knowledge: searching "Times Cited" with Medical Subject Headings (MeSH). *Journal American Society for Information Science and Technology*, 64(5):1076–1080.

Leydesdorff, L., Rotolo, D., and Rafols, I. (2012). Bibliometric perspectives on medical innovation using the Medical Subject Headings of PubMed. *Journal of the American Society for Information Science and Technology*, 63(11):2239–2253.

Lundberg, J., Fransson, A., Brommels, M., Skar, J., and Lundkvist, I. (2006). Is it better or just the same? Article identification strategies impact bibliometric assessments. *Scientometrics*, 66(1):183–197.

Ohniwa, R. L., Hibino, A., and Takeyasu, K. (2010). Trends in research foci in life science fields over the last 30 years monitored by emerging topics. *Scientometrics*, 85(1):111–127.

Rotolo, D., Rafols, I., Hopkins, M., and Leydesdorff, L. (2014). Scientometric mapping as a strategic intelligence tool for the governance of emerging technologies. *Working Paper available at http://arxiv.org/abs/1303.4672*.

Stegmann, J. and Grohmann, G. (2003). Hypothesis generation guided by co-word clustering. *Scientometrics*, 56(1):111–135.

Torvik, V. I., Weeber, M., Swanson, D. R., and Smalheiser, N. R. (2005). A probabilistic similarity metric for Medline records: A model for author name disambiguation. *Journal of the American Society for Information Science and Technology*, 56(2):140–158.

Wickham, H. (2010). stringr: modern, consistent string processing. *R Journal*, 2(December):38–40.

Appendix

Listing 1: *medlineR* script

```
1
2  # medlineR (v.1.0): Matching MEDLINE/PubMed and ISI Web of Science (WoS)
3  # Rotolo and Leydesdorff (2014)
4  # Please refer to the fair use policy at http://wos.isitrial.com/policy/Policy.htm
5
6  # SETTING PARAMETERS - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
7
8  # insert the WoS (MEDLINE interface) link including quotes
9  wosurl<-"..."
10
11 # insert the number of document to match
12 numdocs<-...
13
14 # setting the output folder including quotes
15 # (e.g. "C:\\Users\\user" in Windows or
16 # "/Users/username/Desktop/" in Mac OS X)
17 setwd("...")
18
19 #- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
20
21 # loading library
22 library(stringr)
23
24 # dowloading the html code and parsing data
25 wosurl_str<-substr(wosurl, 1, (nchar(wosurl)-1))
26
27 for(k in 1:numdocs)
28 {
29   print(paste("--Record number: ",k," (out ",numdocs,")--",sep=''))
30   url<-paste(wosurl_str,k,sep='')
31   all_lines<-readLines(url)
32   line<-all_lines[str_detect(all_lines,"UT=WOS:")]
33   line_str<-unlist(strsplit(str_extract(line,"UT=WOS:.+"),"&"))
34   wosut<-sub('WOS:','',line_str[1])
35
36   line<-all_lines[str_detect(all_lines,"NCBI_DB&PMID")]
37   line_str<-unlist(strsplit(str_extract(line,"PMID.+"),"&"))
38   pmid<-line_str[1]
39
40   data<-cbind(pmid,wosut)
41
42   write.table(data,file='wosut.txt',row.names=F,col.names=F,append=T,sep=",")
43   write.table(all_lines,file=paste('wos',pmid,'.txt',sep=''),row.names=F,col.names=F,
         append=F)
44 }
45
46 # creating the search string for WoS
47 uts<-read.csv(file="wosut.txt",header=F,sep=",",fill=T)
48 uts<-subset(uts,uts[,2]!="")
49 searchwos<-paste(uts[,2],collapse=' OR ')
50 write.table(searchwos,file='search.txt',row.names=F,col.names=F,append=F,quote=F)
51 #- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```